

Filed on behalf of EMC Corporation and VMware, Inc.

By: Peter Dichiara, Reg. No. 38,005
David L. Cavanaugh, Reg. No. 36,476
WILMER CUTLER PICKERING HALE AND DORR LLP
peter.dichiara@wilmerhale.com
david.cavanaugh@wilmerhale.com
Tel.: 617-526-6466
Fax: 617-526-5000

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

EMC CORPORATION and VMWARE, INC.,
Petitioners

v.

Patent Owner of
U.S. Patent No. 6,415,280 to Farber et al.

IPR Case No. IPR2013-00083

REPLY DECLARATION OF DOUGLAS W. CLARK, PH.D.

I, Douglas W. Clark, declare as follows:

1. I am the same Douglas W. Clark who submitted a prior declaration in this matter, which I understand was filed on December 15, 2012. My

qualifications remain as stated in paragraphs 1-7 and Appendix A of that declaration, filed as Exhibit 1009 in this case. My statements in paragraphs 8-10 of my prior declaration regarding my review of the '280 patent and related materials also remain unchanged.

2. Since my prior declaration in this case, I have carefully reviewed PersonalWeb's Preliminary Response of March 21, 2013, the Board's Decision to Institute of May 17, 2013, the transcript of my deposition taken on July 10 and July 11, 2013, PersonalWeb's Response of July 24, 2013, the Declaration of Robert B.K. Dewar of July 24, 2013, and the transcript of the deposition of Robert B.K. Dewar taken on September 25 and September 26, 2013.

3. I confirm that everything included in my prior declaration of December 15, 2012, and all of my testimony given during my deposition of July 10 and July 11, 2013 remain true to the best of my knowledge.

Woodhill discloses "data files" that are "named data items"

4. I understand that the Board construed a "data file" as a "named data item, such as a simple file that includes a single, fixed sequence of data bytes or a compound file that includes multiple, fixed sequences of data bytes." (Decision at 11.) Woodhill divides files into one or data streams (Woodhill at 4:13-21; Ex. 1005) and then divides those data streams into one or more binary objects. (Woodhill at 4:21-23; Ex. 1005.) In other words, Woodhill stores "data files

comprised of *one or more* binary objects.” (Woodhill at 2:3 (emphasis added); Ex. 1005.) Accordingly, Woodhill describes “data files” that comprise *one* binary object (i.e., a single, fixed sequence of data bytes) *and* “data files” that comprise multiple binary objects (i.e., multiple, fixed sequences of data bytes.) For each binary object, Woodhill calculates a Binary Object Identifier that is “based on the contents of the binary object so that the Binary Object Identifier 74 changes when the contents of the binary object changes.” (Woodhill at 8:58-62; Ex. 1005.) For files comprising a single binary object, the Binary Object Identifier 74 for that binary object includes a hash of the contents of the entire file.

5. Dr. Dewar notes that “Woodhill defines a ‘file’ as viewed by Distributed Storage Manager program 24 as ‘a collection of data streams,’” and then concludes that “according to Woodhill’s definition, a ‘file’ that is to be backed up or subjected to the auditing procedure has at least two data streams.” (Dewar Decl. at ¶ 108; Ex. 2013.) I disagree with Dr. Dewar’s conclusion. A person of ordinary skill in the art would not have read Woodhill to demand that a file must have at least two data streams. The ordinary meaning of a “collection of items” is a collection that may include zero, one, or many of those items. For example, as confirmed by the ‘280 patent, a file system directory is a collection of files. (‘280 patent at 5:46; Ex. 1001.) A person of ordinary skill in the art would understand that a directory may contain zero, one, or many files. Likewise,

Woodhill's statement that a file can be viewed as "a collection of data streams" (Woodhill at 4:13-15; Ex. 1005) is properly interpreted as meaning that a file has a collection of zero, one or more data streams. Dr. Dewar's reliance on the plural form of "data streams" is misplaced given the ordinary meaning of "collection."

6. Describing a situation where a file has more than one data stream, Woodhill discloses that "a file may contain its normal data and may also contain extended attribute data." (Woodhill at 4:18-19; Ex. 1005.) In this case, such a file would have two data streams: one for its "normal data" and one for its "extended attribute data." However, Woodhill is also clear that its system is "for storing data files comprised of *one or more* binary objects." (Woodhill at 2:2-3 (emphasis added); Ex. 1005.) In the case of a file having only a single binary object, one of ordinary skill in the art would understand that there are four possible cases for this file: (1) the file does not have "normal data" (e.g., it is an empty file), (2) the file does not have "extended attribute data," (3) the file's "normal data" and "extended attribute data" are combined in the file's single binary object, or (4) the file's "extended attribute data" is dealt with in some independent way. In each of these cases, the binary object for that file stores all of the file's data. Thus, I disagree with Dr. Dewar's conclusion that "[e]ven if a 'file' in Woodhill were to include only one 'binary object', this does not necessarily mean that the binary object makes up the entire file." (Dewar Decl. at ¶ 107; Ex. 2013.)

7. Furthermore, I point out that Woodhill discloses a File Identification Record 34 that is saved for each data file that is backed up. (Woodhill at 3:54-57; Ex. 1005.) Each such record includes a File Name 40 field, which is the “name of the file.” (Woodhill at 5:61; Ex. 1005.) Thus, each of Woodhill’s data files is a “named data item.” While Dr. Dewar confirms that “File names 40 in Woodhill identify files,” he goes on to state that Woodhill’s File Name 40 fields do not identify binary objects (Dewar Decl. at ¶ 105; Ex. 2013). I disagree that Woodhill’s File Names 40 do not identify binary objects for cases of files having only a single binary object. As described above, Woodhill discloses files having only one binary object. Therefore, for files that have only one binary object, a person of ordinary skill in the art would understand that the File Name 40 field names the single binary object for that file. Woodhill further discloses the case where a “calculated binary object identifier [is] saved as the name of the associated binary object” (Woodhill at 22:3-4; Ex. 1005), and so binary objects are further named by their Binary Object Identifiers.

Woodhill requests “data files” using Binary Object Identifiers

8. I describe below two examples of client requests described by Woodhill for restoring data files: (1) self-audit requests for a randomly selected binary object and (2) update requests to restore a granularized file to a previous version of that file. As described in more detail below, a self-audit request is a

request for a randomly-selected binary object, the request including the Binary Object Identifier for that binary object. For files having only a single binary object, this request restores the entire contents of the file in response to a request that includes a hash of the entire contents of the file. Also, as described in more detail below, an update request is a request to restore a current version of a file to a prior version of that file. In the case where the current version of a file has significantly changed from the previous version of that file, the entire contents of the file may be transmitted to the requesting local computer. This scenario may occur if, for example, a disk failure or virus on a local computer destroys or overwrites each binary object of a particular file.

9. A person of ordinary skill in the art would understand, however, that these are two specific implementations of generally well-known restore techniques. Indeed, Woodhill notes that “[b]ackup/restore systems have a long history on all types of computer systems” (Woodhill at 1:23-24; Ex. 1005) and describes its invention as being “for the management of the storage space on a computer system [that] provide[s] a backup/restore system.” (Woodhill at 2:40-41; Ex. 1005.) Accordingly, while Woodhill explicitly describes the implementation of its self-audit request mechanism (*See* Woodhill at 18:10-38; Ex. 1005), the purpose of this mechanism is “to ensure that the binary objects that have been backed up can be restored.” (Woodhill at 18:12-13; Ex. 1005.) In other words, the self-audit request

mechanism ensures, *before* a disk failure or a disaster strikes, that it will be possible to restore files *after* such an event. Indeed, the very goal of Woodhill's backup procedure is to "ensure[] that at least one copy of every binary object is stored and that a disaster that destroys an entire site would not destroy all copies of that site's data." (Woodhill at 9:42-44; Ex. 1005.) Thus, it is clear that Woodhill includes the ability to restore files previously backed up. Rather than explain in detail the general case of such a well-known procedure, Woodhill describes the implementation of its more complicated "update request" procedure for restoring *granularized* binary objects. (See Woodhill at 17:18-18:9; Ex. 1005.) Because (as described below) an "update request" includes Binary Object Identifiers for the binary objects of a file to be restored *and* current granule contents identifiers (Woodhill at 17:42-46; Ex. 1005), it is clear that the more general case of restoring a non-granularized file would include Binary Object Identifiers for the binary objects of that file. The only difference between these cases is that granule contents identifiers would not be necessary for non-granularized file. One skilled in the art would appreciate that a "back-up" writes a version of a file to the back-up server, and that a "restore" reads a version of a file from the back-up server. These are well-understood data management techniques that existed long before the patents.

10. Despite the existence of well-known mechanisms for restoring data files, Dr. Dewar argues that “Woodhill fails to disclose a client request for a data file, where the ‘request’ includes ‘a hash of the contents of the data file.’” (Dewar Decl. at ¶ 102; Ex. 2013.) In particular, Dr. Dewar argues that, “in Woodhill’s self-auditing procedure there is no description of a ‘request’ for a binary object that includes [Binary Object Identification] Record 58.” (Dewar Decl. at ¶ 104; Ex. 2013.) However, as I described in my prior declaration for this case, Woodhill indeed requests binary objects using the Binary Object Identifier 74 portion of the corresponding Binary Object Identification Record 58 during its self-audit procedure. For example, during Woodhill’s self-audit procedure, the “Distributed Storage Manager program 24 initiates a restore of a randomly selected binary object identified by a Binary Object Identification Record 58 stored in File Database 25.” (Woodhill at 18:17-19; Ex. 1005.) A person of ordinary skill in the art would understand this sentence to mean that Woodhill randomly selects a Binary Object Identification Record 58 from the potentially many such records stored in File Database 25, and requests the binary object corresponding to that record. For such a routine operation, there was no need for Woodhill to explicitly describe that the binary object identifier from the record is included as part of a request. Rather, it was well-known at the time of the Woodhill patent how to request objects using their identifiers. In my prior declaration for this case, I

mentioned many such examples, including the Langer reference (which dates back to 1991). PersonalWeb seems to have conceded this point by noting that “Langer appears to disclose accessing a standalone file by employing an MD5 of the file contents.” (IPR2013-00085, Resp. 41; Ex. 1079.) Accordingly, a person of ordinary skill in the art would understand that Woodhill’s remote backup file server can provide binary objects in response to their identifiers. Woodhill’s Figure 3 makes clear that the Binary Object Identifier 74 portion of the Binary Object Identification Record 58 is the key part of that record. (Woodhill at Fig. 3; Ex. 1005.)

11. During Woodhill’s self-audit procedure, the binary object is “identified by a Binary Object Identification Record 58.” (Woodhill at 18:18-19; Ex. 1005.) As described above, a person of ordinary skill in the art would understand that the key component of this record is the Binary Object Identifier 74. (Woodhill at Fig. 3; Ex. 1005.) However, rather than acknowledge the role of Binary Object Identifier 74, Dr. Dewar lays out a sequence of steps using the other components of the Binary Object Identification Record 58, and argues that “[t]he ‘Link to Backup Instance Record 60’ field is the field used in Woodhill’s self-auditing procedure to reach the ‘Link to File Identification Record 44.’ And the ‘Link to File Identification Record 44’ is then used to reach ‘File Identification Record 34.’ The ‘File Name 40’ and ‘File Location 38’ fields in the ‘File

Identification Record 34' are used to access the file containing the binary object at backup storage. Then, the "Binary Object Stream Type 62" and "Binary Object Offset 72" are used to locate the binary object in that file." (Dewar Decl. at ¶ 53; Ex. 2013.) Because Woodhill's self-audit procedure is capable of restoring a binary object stored on the remote backup file server, it would not make sense to use this sequence of steps to access a binary object. First, the Binary Object Stream Type field 62 and the Binary Object Offset field 72 would not be useful in locating a binary object, as these fields do not identify any particular binary object. Second, the Link to Backup Instance Record 60 field links to another record stored in the File Database, which is a database stored and maintained on a local computer. (Woodhill at 3:45-47; Ex. 1005.) Thus, the linked record would not be available on the remote backup file server, unless the corresponding Backup Instance Record was also sent from the local computer to the remote backup file server (which would not be efficient, and which Woodhill does not suggest in any event). Furthermore, Dr. Dewar suggests using not just one link (the Link to Backup Instance Record 60), but a second link (the Link to File Identification Record 44) to access yet another record. (Dewar Decl. at ¶ 53; Ex. 2013.) Again, these records are all stored on a local computer's File Database, and so it would not make sense that they would be used to access a binary object from the remote backup file server in this way. Even if these fields were accessible by, or

transmitted to, the remote backup file server, the File Name 40 and File Location 38 fields represent the local location of the file on a local computer, not on the remote backup file server. Thus, they would not be useful in accessing a binary object on the remote backup file server. Put more simply, the fields of a Binary Object Identification Record 58, other than the Binary Object Identifier 74 portion, are simply not relevant to accessing a binary object on the remote backup file server.

12. I also point out that Woodhill's self-audit procedure is clearly initiated by a local computer. Woodhill describes that, "the Distributed Storage Manager program 24 initiates a restore of a randomly selected binary object identified by a Binary Object Identification Record 58 stored in File Database 25." (Woodhill at 18:16-19 (emphasis added); Ex. 1005.) Woodhill is clear that its "discussion illustrates the operation of the Distributed Storage Manager program 24 on a single local computer 20" and that "it should be understood that the Distributed Storage Manager program 24 operates in the same fashion on each local computer 20." (Woodhill at 5:4-8; Ex. 1005.) Therefore, one of ordinary skill in the art would understand that the self-audit procedure is clearly initiated by the Distributed Storage Manager program 24 as it executes on a local computer. This is further confirmed by Woodhill's reference to initiating the self-audit using "a Binary Object Identification Record 58 stored in File Database 25." (Woodhill at 18:18-

19 (emphasis added); Ex. 1005.) Woodhill is clear that “[t]he Distributed Storage Manager program 24 of the present invention builds and maintains the File Database 25 on one of the disk drives 19 on each local computer 20” (Woodhill at 3:45-47; Ex. 1005), and so the self-audit procedure initiates the self-audit using the Binary Object Identification Record 58 from the File Database 25 on the local computer. As described above, the Binary Object Identifier 74 portion of this record is then transmitted to the remote backup file server. Each Binary Object Identifier 74 includes a Binary Object Hash 70 field. (Woodhill at Fig. 3; Ex. 1005.) Accordingly, when a local computer initiates the self-audit procedure, a binary object is requested using a hash of the contents of the binary object. In the case of a data file having a single binary object, a self-audit request for that binary object is a request for that data file. A self-audit request includes a hash of the contents of the binary object, which, in this case, is a hash of the contents of the entire data file.

13. Because Woodhill requests binary objects using their Binary Object Identifiers, I also disagree with Dr. Dewar’s conclusion that Woodhill’s Binary Object Identifiers are “simply used for comparison purposes after the binary object has already been accessed.” (Dewar Decl. at ¶ 103; Ex. 2013.) Woodhill’s Binary Object Identifiers are called “identifiers” precisely because they are used for more than comparison. As described above, Binary Object Identifiers are used for

requesting binary objects and Woodhill's claim 1 even makes clear that a binary object identifier is "saved as the name of the associated binary object." (Woodhill at 22:3-4; Ex. 1005.)

14. Furthermore, in the more general case of a single binary object being requested where that binary object belongs to a file comprising many binary objects (i.e., a compound file), Woodhill works in the same way as the '280 patent. For a read operation involving a compound file, the '280 patent's Read File mechanism "break[s] the read operation into one or more read operations on component segments" by "[i]dentify[ing] the segment(s) to be read" and "[u]s[ing] the Read File mechanism (recursively) to read from [each] segment." ('280 patent at 21:30-50; Ex. 1001.) Similarly, the '280 patent makes clear that its "Request True File remote mechanism copies only a single data item from one processor to another. If the data item is a compound file, its component segments are not copied, only the indirect block is copied. The segments are copied only when they are read (or otherwise needed)." ('280 patent at 34:4-8; Ex. 1001.) In both of these cases, requests for compound files do not cause the underlying file data to be retrieved. Instead, requests including compound file hashes merely retrieve an indirect block of segment hashes. Segment data is only provided in response to subsequent requests that include those segment hashes. Like the '280 patent's requests for data of a single segment of a compound file, Woodhill's self-audit

request may retrieve a single binary object from a file comprising multiple binary objects.

15. As another example of requesting a data file using Binary Object Identifiers 74, Woodhill discloses an “update request” that is transmitted to the remote backup file server. (Woodhill at 17:41-43; Ex. 1005.) The update request is used to “update” a file as it currently exists on a local computer to a version that was previously backed up. (See Woodhill at 17:18-43; Ex 1005.) The update request works for files, such as large database files, that were previously backed up using Woodhill’s “granularization” procedure. (Woodhill at 14:53-65; Ex. 1005.) Woodhill is explicit that each update request “includes the Binary Object Identification Record 58 for the previous version of each binary object.” (Woodhill at 17:41-45; Ex. 1005.) Each of these Binary Object Identification Records 58 includes a Binary Object Identifier 74 portion and each Binary Object Identifier 74 includes a Binary Object Hash 70 value. (Woodhill at Fig. 3; Ex. 1005.) Therefore, an update request includes a hash of the contents of the file being requested. Also included in an “update request” are “‘contents identifiers’ for each ‘granule’ within the current version of each binary object as it exists on the local computer 20.” (Woodhill at 17:41-47; Ex. 1005.) Woodhill’s remote backup file server proceeds to compare the granule contents identifiers for the current version of each binary object with granule contents identifiers

corresponding to the requested previous version of those binary objects. (Woodhill at 17:50-55; Ex. 1005.) In each case, when the contents identifiers “do not match, . . . the Distributed Storage Manager program 24 transmits the ‘granule’ to the local computer 20. . . . After all ‘granules’ are received from the remote backup file server 12, the binary object has been restored to the state of the previous version.” (Woodhill at 17:62-18:9; Ex. 1005.) Therefore, for any of the requested previous versions of the binary objects, each granule that differs from the version as it existed on the local computer (at the time of the request) is sent to that local computer. Accordingly, if every granule of a previous version of a binary object differed from the corresponding current granule, every granule of that binary object would be transmitted from the remote backup file server to the requesting local computer. In this way, the full contents of a previous version of a binary object would be restored in response to a request including the Binary Object Identifier 74 of that binary object. Therefore, the full contents of a previous version of a file may be transmitted from the remote backup file server to the requesting local computer in this way (*e.g.*, if every granule of every binary object for that file differed from the corresponding granule that was previously backed up). Even if some granules are the same, the full contents of a file are nonetheless restored by the update request mechanism.

Woodhill backs up files regardless of their number of binary objects

16. Dr. Dewar argues that “Woodhill’s definition of ‘file’ ...requires that each ‘file’ to be backed up includes at least two binary objects.” (Dewar Decl. at ¶ 108; Ex. 2013.) However, explained above, Woodhill clearly discloses files having only a single binary object. Furthermore, Woodhill in no way suggests that files with only a single binary object should be treated any differently than files having two or more binary objects. In fact, if Woodhill implemented an arbitrary distinction that prevented files with only a single binary object from being backed up or audited, system performance could dramatically suffer. For example, a document may be very small (and thus fit within a single binary object), yet nonetheless be very important. For example, a resume or a copy of a lease may be only a few pages long (and thus likely fit well within a single binary object), while still being one of the most important documents stored on a user’s computer. A user would expect that backup copies of these important documents (regardless of their number of binary objects) would be stored so that they could be safely recovered if a disaster destroyed the user’s computer or damaged its disk drives. Woodhill’s system is designed to “ensure[] that at least one copy of every binary object is stored and that a disaster that destroys an entire site would not destroy all copies of that site’s data.” (Woodhill at 9:42-44; Ex. 1005.) Likewise, the self-audit procedure “ensure[s] that the binary objects that have been backed up can be

restored.” (Woodhill at 18:12-13; Ex. 1005.) If Woodhill only backed up or self-audited files having two or more binary objects (as Dr. Dewar has claimed), then Woodhill’s system would not effectively prevent a disaster from destroying all copies of those files having only a single binary object. Therefore, a person of ordinary skill in the art would not read Woodhill to demand that files to be backed up or subject to the auditing procedure are in any way distinct from files otherwise described by Woodhill.

Availability for Cross-Examination

17. In signing this declaration, I recognize that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross examination in the case and that cross examination will take place within the United States. If cross examination is required of me, I will appear for cross examination within the United States during the time allotted for cross examination.

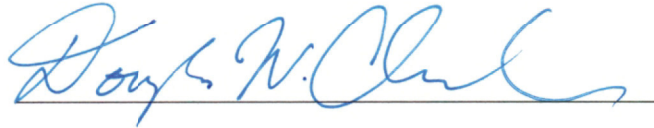
Right to Supplement

18. I reserve the right to supplement my opinions in the future to respond to any arguments that Patentee raises and to take into account new information as it becomes available to me.

Jurat

19. I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: October 1, 2013

A handwritten signature in blue ink, appearing to read "Douglas W. Clark", written over a horizontal line.

Douglas W. Clark
Philadelphia, PA